Потоковые симметричные шифры Защита информации

Колыбельников Александр

Московский физико-технический институт
(государственный университет)

29 сентября 2021 г.

Потоковым или поточным называется шифр в котором каждый символ открытого текста преобразуется в символ шифрованного текста в зависимости не только от
используемого ключа, но и от его расположения в потоке открытого текста.

Пример потокового шифра побитное сложение потока данных $C_i = K_i \oplus M_i$. Объектом обработки в потоковом шифре может быть:

- бит;
- байт;
- слово.

Из теории Шеннона существуют несколько выводов справедливых для потоковых шифров. Для того, что бы потоковый шифр был абсолютно стойким должны выполнятся следующие условия:

- Длина ключа потокового шифра $L_k \geq L_m$;
- Биты ключа потокового шифра должен быть равновероятен и независим;
- Открытый текст M и шифртекст C должны быть независимы

$$H(M|C) = H(M), I(M; C) = 0.$$

Требования к современным потоковым шифрам были сформулированы Ади Шамиром для европейского конкурса поточных шифрова E-stream.

- Аппаратная реализация потокового шифра должна быть быстрее блочного шифра;
- Аппаратная реализация потокового шифра не должна требовать вычислительных ресурсов больше чем блочный шифр;
- Программная реализация потокового шифра должна быть быстрее блокового шифра;

Потоковые шифры бывают нескольких типов:

- Синхронные – $l_k = l_m$, биты зашифрованного текста между собой независимы;
- Самосинхронизирующиеся(асинхронные) – $l_k > l_m$, биты зашифрованного текста зависят между собой.

Положительные свойства

- отсутствие эффекта распространения ошибок;
- обнаруживают вставки и удаления шифртекста.

Отрицательные свойства

- уязвимы для изменения отдельных бит шифрованного текста. Если злоумышленнику известен открытый текст, он может изменить эти биты так, чтобы они расшифровывались, как ему надо.

Положительные свойства

- каждый знак открытого текста влияет на следующий шифротекст, статистические

свойства открытого текста распространяются на весь шифротекст.

Отрицательные свойства

- распространение ошибки, одна ошибка искажает весь текст за ней;
- не стойкие к атаке повторной передачи

сообщения.

Потоковые шифры бывают нескольких видов:

- Сумматор потока битов открытого текста с ключом $\oplus$;
- Сумматор потока байтов открытого текста с ключом – .

- Комбинации из сумматоров.

В этом случае не сложно заметить, что требование по скорости работы в первую очередь относится к генератору ключей.

Алгоритм был предложен Лемером в 1949 году.

$$x_{n+1} = a \cdot x_n + c \bmod m.$$

Числа a, c, m, $0 < a < m$, $0 < c < m$, являются параметрами алгоритма.
Максимальный период ограничен значением m.
Максимум достигается при условии

- числа c и m взаимно просты;
- число $a - 1$ кратно каждому простому делителю числа m;
- число $a - 1$ кратно 4, если m кратно 4.

Зная два последовательных значения выхода генератора (x_n и x_{n+1}) и единственный параметр схемы m, можно

решить систему уравнений и найти a и c, чего будет достаточно для нахождения всей дальнейшей (или предыдущей) части последовательности. Параметр m, в свою очередь, можно найти перебором, начиная с некоторого $\min(X) : X \geq x_i$, где x_i – наблюдаемые элементы последовательности.

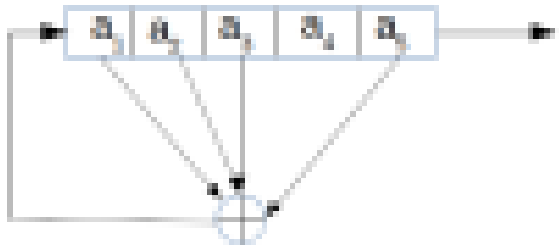
Для параметров a = 2, c = 3, m = 5 и начального

состояния $x_0 = 1$ получаем последовательность: 0, 3, 4, 1, 0, . . .

Привести следующие два элемента последовательности, сформированной линейным конгруэнтным методом, если предыдущие 3 элемента последовательности такие:
268, 411, 253, а все вычисления

выполняются в поле F_{499}. Колыбельников Александр

Линейный рекурентный регистр сдвига, он же регистр сдвига с линейной обратной связью (РСЛОС) в самом упрощенном виде выглядит следующим образом.



Начальным состоянием генератора является набор значений в битовых ячейках. На каждой итерации генератор вычисляет сумму по модулю два (то есть выполняет операцию XOR) значений ячеек, для которых $C_i = 1$:

$$b_{n+1} = \sum_i C_i b_i \bmod 2,$$

$$b_{n+1} = b_1 \oplus C_2 b_2 \oplus C_3 b_3 \oplus \cdots \oplus C_n b_n.$$

Далее регистр сдвигает значения на одну ячейку влево. Самая правая ячейка b_n принимает вычисленное значение b_{n+1}:

$$b_1 := b_2,$$
$$b_2 := b_3,$$
$$\cdots$$
$$b_n := b_{n+1}.$$

Выход генератора – значение ячейки b_1 после сдвига.

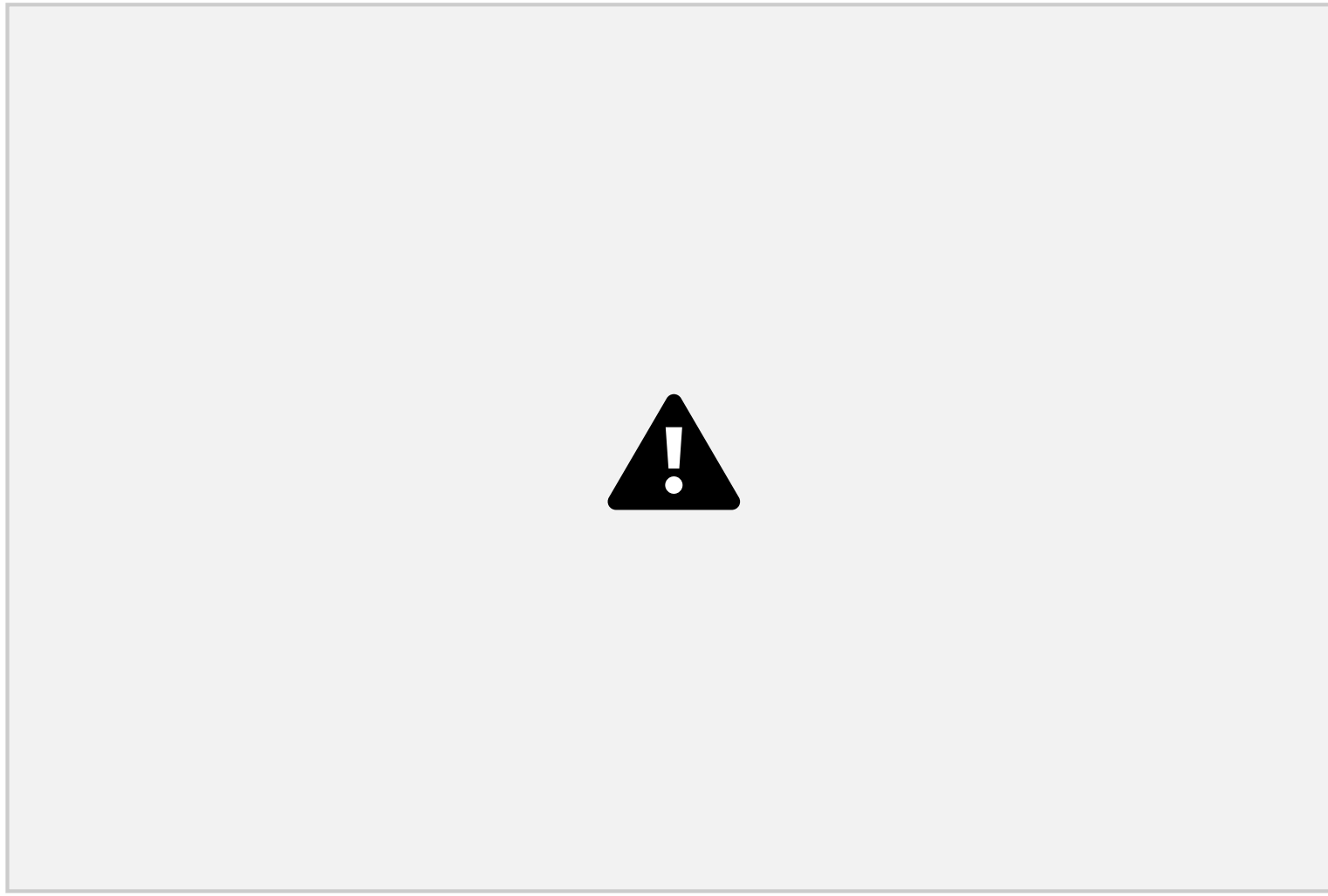
Важнейшим свойством РСЛОС, влияющим на структурную стойкость является период генератора. Максимальный период последовательности РСЛОС равен $2^n - 1$. Максимум достигается в том и только в том случае, когда

характеристический многочлен РСЛОС примитивен.
Период генератора определяет через какое количество бит выход генератора начнет повторяться.
Если известна структура РСЛОС (значения коэффициентов $C_2, \ldots, C_n$), то внутреннее состояние генератора можно восстановить по n предыдущим выходам. По 2n предыдущим выходам генератора можно восстановить и внутреннее состояние, и структуру генератора. Зная структуру и текущее внутреннее состояние генератора, можно
восстановить его предыдущие и следующие выходные значения.

Приведите предыдущие 5 бит выхода генератора псевдослучайной последовательности, основанного на

регистре сдвига с линейной обратной связью, если известно, что характеристический полином регистра — $m(x) = x^5 + x^3 + x^2 + x + 1$, а дальнейшая последовательность такова: 1, 0, 0, 0, 1, 0. Генератор приведен на рисунке.

Рис.: Генератор Blum-Blum-Shub

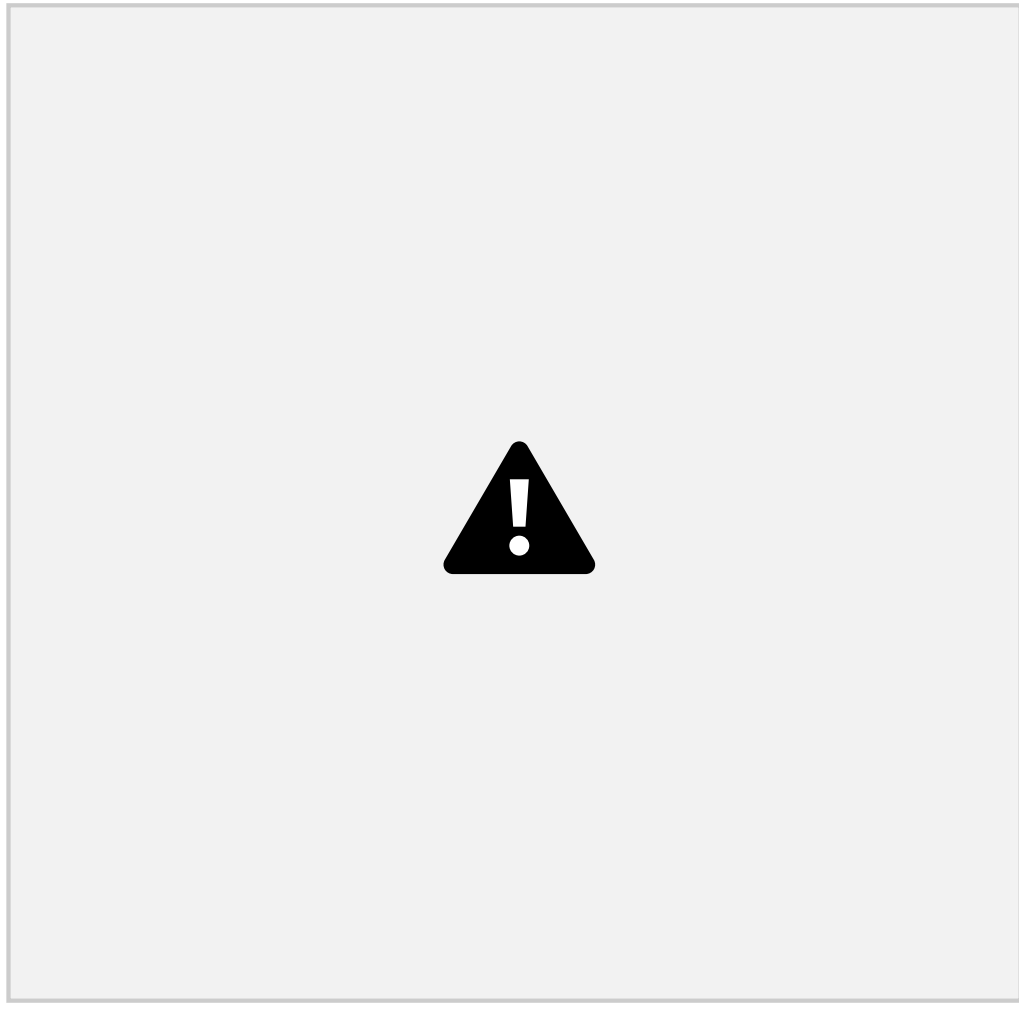


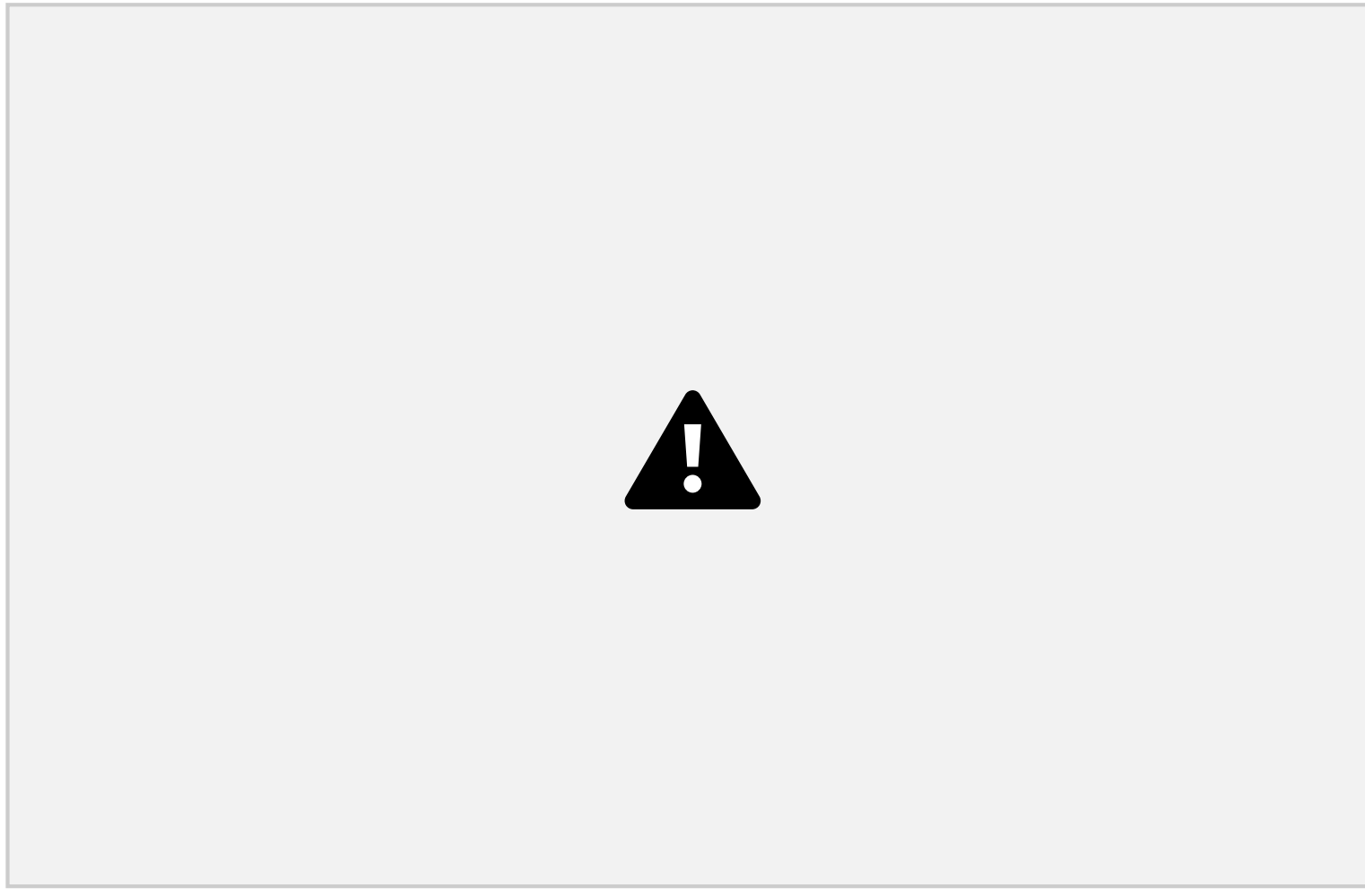
Задача алгоритма – защита от клонирования телефона

путем генерации отзыва Signed Response на случайный пароль (RAND – Random), получаемый сотовым телефоном (Mobile Station) от центра коммутации Mobile Switching Centre в процедуре аутентификации.

1. x[16-31] = RAND
2. for 0 < i < 8

   x[0 − 15] = K_i
   call Compression (5 rounds)
   call FormBitsFromBytes
   if i<7 call Permute





Алгоритм A8 вычисляет ключ шифрования Kc из случайной

последовательности RAND получаемой при процедуре аутентификации, с использованием ключа аутентификации K_i.

- длина K_i;: 128 бит;
- длина RAND: 128 бит;
- длина K_c: 64 бит.

- Системно-теоретический подход основан на создании для криптоаналитика сложной, ранее неисследованной проблемы.
- Сложностно-теоретический подход основан на сложной, но известной проблеме.
- Информационно-технический подход основан на попытке утаить открытый текст от криптоаналитика.
- Рандомизированный подход основан на создании вычислительно сложной задачи.

- длинные периоды выходных последовательностей;
- диффузия – рассеивание избыточности в подструктурах, «размазывание» статистики по всему тексту;
- каждый бит потока ключей должен быть сложным преобразованием большинства битов ключа;
- критерий нелинейности для логических функций;
- большая линейная сложность.

1. Нахождение периода генератора;
2. Слабые ключи;
3. Time-memory-tradeoff атака;
4. Поиск(измерение) равнораспространений;
5. Корреляционная атака;
6. Дифференциальная атака;
7. Алгебраические атаки;
8. Атака на восстановление ключа и вектора IV.